MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS-1963-A

Report 2387

ROBOTIC TECHNOLOGY APPLIED TO ARMY MOBILITY SYSTEMS

by
Reinaldo J. Chavez
Charles A. Amazeen
Harry L. Keller

July 1983

DTIC
ELECTE
NOV 1 5 1983

S D

A

**U.S. ARMY MOBILITY EQUIPMENT
RESEARCH AND DEVELOPMENT COMMAND
FORT BELVOIR, VIRGINIA**

MERADCOM
U.S. ARMY MOBILITY EQUIPMENT
RESEARCH & DEVELOPMENT COMMAND

DTIC FILE COPY

83   11   15   055

Destroy this report when it is no longer needed.
Do not return it to the originator.

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 2387	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) ROBOTIC TECHNOLOGY APPLIED TO ARMY MOBILITY SYSTEMS		5. TYPE OF REPORT & PERIOD COVERED MERADCOM Tech Report Aug 82–Jun 83
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Reinaldo J. Chavez Charles A. Amazeen Harry L. Keller		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Combined Arms Sup Lab; ATTN: DRDME-XS; US Army Mobility Equipment Research and Development Command; Fort Belvoir, VA 22060		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62733A 1116733AH20CT
11. CONTROLLING OFFICE NAME AND ADDRESS Combined Arms Sup Lab; ATTN: DRDME-XS; US Army Mobility Equipment Research and Development Command; Fort Belvoir, VA 22060		12. REPORT DATE July 1983
		13. NUMBER OF PAGES 58
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Robotics	Sensors
Microprocessor Control	Control Theory
Robotic Technology	

backhoe

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes the system design and modifications associated with the implementation of a hierarchical 8-bit microprocessor control structure to the excavation cycle of a J. I. Case Model 35 backhoe.

The feedback control structure associated with driving a third order system consisting of electro-hydraulic valves, microprocessors, optical encoders, and inertial loads is described along with an analysis of the predicted transient response.

PREFACE

Accession For

NTIS GRA&I ☑

DTIC TAB ☐

Unannounced ☐

Justification

Distribution/

Availability Codes

Avail and/or

Dist Special

DTIC COPY INSPECTED 2

CONTENTS

ILLUSTRATIONS

ROBOTIC TECHNOLOGY APPLIED TO ARMY MOBILITY SYSTEMS

I. INTRODUCTION

Recognizing that the U.S. Army Corps of Engineers has a need for applying industrial robotic technology to high-risk and labor intensive tasks performed in the environment of a modern battlefield, USAMERADCOM started a small investigative project in a specific robotic application area—robotics applied to rapid excavation. The excavator used was a modified J. I. Case Model 35 Backhoe mounted on a highly mobile Mercedes Unimog truck.

The conditions of the modern battlefield make rapid excavation extremely hazardous to the human operators due to the advent of highly accurate, technologically sophisticated weapons systems and the possibility of nuclear, biological, and/or chemical contamination of the battlefield environment.

The goal, then, of the project was to eliminate or reduce through automation as much of the manual digging operation of the backhoe as would be consistent with current operating procedures. In the process of doing this, the technological expertise developed in the area of microprocessors, sensors, and servo systems would be applied to a broad spectrum of MERADCOM mission areas.

With this goal in mind, what will be presented in this report will be the preliminary results associated with controlling the backhoe arm through a series of sequential motions. The complete cycle—controlling program will be discussed only from a conceptual viewpoint while the more primitive state driving and state sensing programs are explained in detail.

II. HIERARCHICAL CONTROL SYSTEM

The microprocessor control system consists of three separate central processing units (CPUs) of the 6502 type. There are two levels of control associated with the system. The first, or primitive level, consists of those programs which determine the state or current position of the backhoe arm (bucket, dipper, boom, and swing) and of the calculations and diagnostics associated with producing the necessary voltages that drive the proportionally controlled electro-hydraulic servo valves. (See Figure 1.)

The higher level consists of a single overall system control program which interacts with the human operator in a real time basis using a "menu" driven approach. From the menu, one of three operational modes can be chosen. The backhoe can be entirely controlled by the operator using joysticks, or the operator can choose a hole predefined in the control program, or the operator can input the dimensions of a hole to be excavated. These operation modes can be found in the program as subroutines or, as called for in PASCAL,[1] procedures. The microprocessors selected were two Rockwell AIM 65s and one Apple II system. This equipment was used due to availability and budgetary constraints.

---

[1] The higher level control program is written in UCSD PASCAL and utilizes on-line floppy disks that store the segmented procedures.
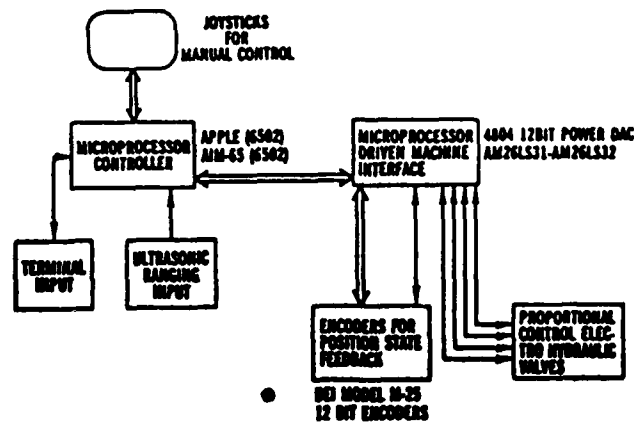
Figure 1. Control hierarchy.

III. SYSTEM CONTROL PROGRAM

The system control program is the software which will operate the excavator while running in the Apple II and interfaced by an RS 232 serial connection to the subordinate AIM 65 processors which will maintain the servo loops.

The high-level functions necessary to operate in the hole template mode have been run using a graphic kinematic simulation to observe results and have operated successfully in that environment.

The total controller (to include the template mode) is approximately 90 percent complete in detailed code at this time; two parts which require some additional code generation before testing are the bucket loading cycle procedure and the controller-to-subordinate processor driver. This program uses PASCAL as its implementation language except for a few assembly language I/O drivers (Analog-to-digital converter and serial interface drivers). The initial draft of detailed code is expected to be ready for evolutionary testing in the fall of 1983, after checkout of the four servo loop subsystems under control of AIM 65.1 and AIM 65.2 is completed. (See Appendix B: Program Control.)

2

IV. STATE DETERMINATION

In the state sensing portion of the state determination program, the state vector is uniquely defined by the four angles α, β, σ, and ϕ. These angles are determined by the digital (10-bit) output of the optical encoders mounted on the bucket, dipper, boom, and swing assembly of the backhoe. (See Figure 2.)

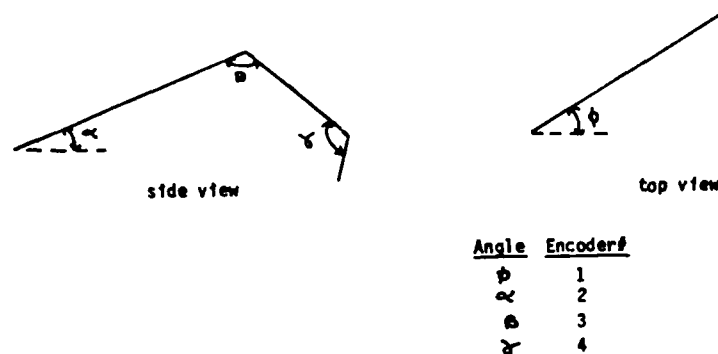| Angle	Encoder#
ϕ	1
α	2
β	3
γ	4

Figure 2. State determination.

The state determination program developed uses the 6502 CPU Assembly Language instruction set to control both the data bus (AM26LS31 - AM26LS32) and the encoder select enable lines.[2] (See Appendix A.)

Basically, the state determination program manages or strobes the optical encoder's 10-bit word onto the data bus, such that only one 10-bit word is on the bus line per decoded encoder enable. This sequence prevents any type of uncertainty on the data bus from occurring. It is, of course, apparent that the RS422 tri-state buses (uses AM26LS31 — AM26LS32) is enabled or in the transmit mode continuously. By continuously enabling this parallel bus, each encoder is enabled on to the bus for transmission of data to take place. The main advantage is one of using less hardware; namely, one RS422 parallel bus services as many devices as are strobed.

---

[2] A. Osborne. "6502 Assembly Language Programming." McGraw-Hill. pp. 25-30. 1980.

3

The enabling sequence is program controlled by the state determination program[3] using 8-bit words written onto the output registers of the AIM 65s' Versatile Interface Adapter (VIA). The 6522 VIA is an enhanced version of the 6502 Peripheral Interface Adapter (PIA).[4] The main difference between the VIA and the PIA is that the VIA provides all the functions of the PIA plus 2-bit counter/timers and an 8-bit shift register. The shift register is a particularly useful addition since it allows for fast transfer of serial data between the two AIM 65s without a special bus interface. This area will be described in more detail in the next section.

There are only 2 bits used of the 8-bit word sent to the output registers of the VIA for the encoder enable. The actual optical encoder is generated by a demultiplexed/decoded high output generated for a SN74156/SN7404 encoder select circuit.

The AIM 65 applications connector which ties in directly to the applications VIA, provides pins PB2 for the coded signal that is used by the demultiplexer/decoder to output lows to the HEX inverters. As previously stated, PB3 and PB2 are 2 bits of an 8-bit word sent under program control to the VIA. The actual assignment of the control words bits as either input or output ports of the VIA is done utilizing the memory mapped registers provided to the AIM's Central Processing Unit (CPU) by the applications VIA. See the state determination program for the actual HEX address.

Once the 10-bit optical encoder is selected into the VIA pins PA0-PA7 and PA0-PB2, then a transfer of their values is done using the accumulation command LDA from the HEX address that has the memory mapped I/O port. At this point, trivial calculations can be made to determine that state vector or the state of the backhoe. The complete flow chart showing the state determination program as well as both a basic driven encoder select program and an assembly language encoder select program is shown at Figure 3.

---

[3] Manual "R6500 Microcomputer System Hardware," Rockwell International, pp. 5-14-6-5, 1980.

[4] Ibid.
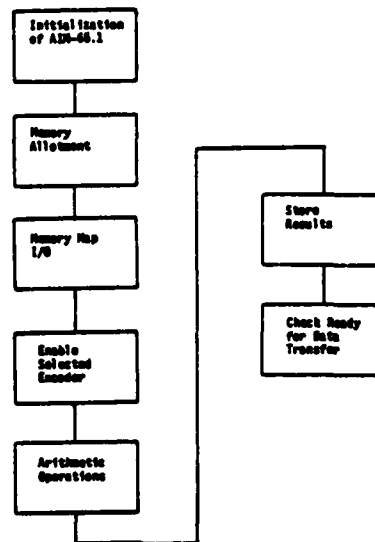
```
K * = 300
/43
0300 A9 LDA #00        032E 85 STA 33
0302 8D STA A003       0330 A9 LDA #10
0300 A9 LDA #FC        0331 8D STA A000
0307 8D STA A002       0335 AD LDA A001
030A AD LDA #04        0338 85 STA 34
030C 8D STA A000       033A AD LDA A000
030F AD LDA A001       033D 18 CLC
0312 85 STA 30         033E 38 SEC
0214 AD LDA A000       033F E9 SBC #10
0317 18 CLC            0341 85 STA 35
0318 38 SEC            0343 A9 LDA #30
0319 E9 SBC #04        0345 8D STA A000
031B 85 STA 31         0348 AD LDA A001
031D A9 LDA #00        0328 85 STA #36
031F 8D STA A000       0340 AD LDA A000
0322 AD LDA A001       0350 18 CLC
0325 85 STA 32         0351 38 SEC
0327 AD LDA A000       0352 E9 SBC #30
032A 18 CLC            0354 85 STA 37
032B 38 SEC            0356 00 BRK
032C E9 SBC #08        0357 00 BRK
                       0360 00 BRK
```

Figure 3. Assembly language encoder select program and flowchart.

Once the state vector calculations are made,[5] if required, then the Matrix Transfer program is used. Specifically, what is denoted by Matrix is the unique set of four 10-bit words which define a point in configuration space as a location point for the final bucket position. The transfer portion is initiated upon either a request from the Apple microprocessor or the AIM 65.2 microprocessor. The requests from the Apple microprocessor are done under separate program control and are periodic system status updates. The actual request is received from the Apple by the AIM 65.1 utilizing a RS232C port controlled by two 6850 an Asynchronous Communications Interface Adapters (ACIA).[6] The request from the AIM 65.2 is received and enabled utilizing the shift registers of VIAs and the control lines CB1, 2 and PB4, 5.

The RS232C Protocol[7] used is essentially one of first determining the Data Terminal (DTR) device and the Data Set (DSR) device. In this application, the DSR device is the Apple microprocessor and the DTR device, the AIM 65 microprocessor. The driving rationale for this decision is one of minimizing the hardware modifications to either microprocessor.

Once the DTR and DSR have been selected, the standard sequence for communication is initiated. (See Figure 4.)

---

[5] J. F. Wakerly, "Microcomputer Architecture and Programming," Wiley, pp. 251-252, 1981.

[6] Manual "Model 7710 ACIA," California Computer Systems, pp. 4-2-4-11, 1981.

[7] Manual "Data Communications Interface-4051," Tektronix, Inc., pp. 2-4-2-9, 1981.
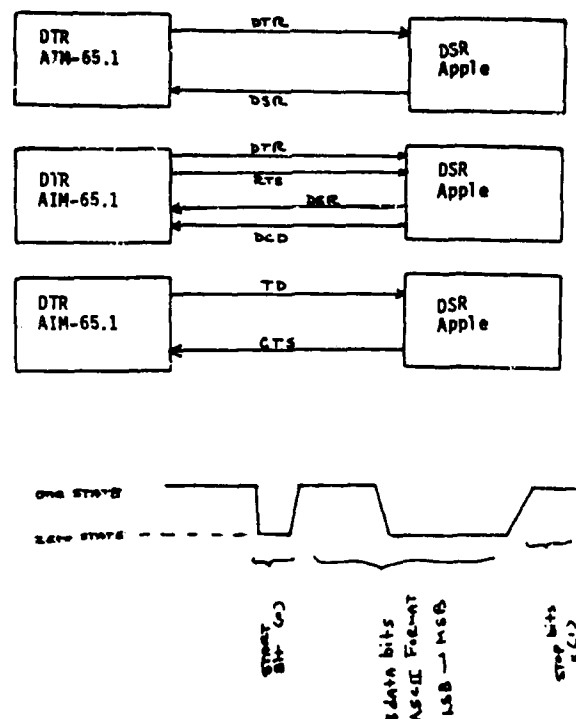
Figure 4. RS232C protocols.

The communication rate has been selected as 9600 baud with 8-bit words, two stop bits, one start bit, and no parity check. The framing and bit width are all being taken care of by the on-board ACIAs.

The configuration of the start, data, and stop bits are shown in Figure 4. The bit width calculation is based upon 11 bits transmitted per frame; e.g.,

Frame Width Calculation

CPS = Characters per second

$$9600 \text{ baud} = Y \text{ (CPS) X } 11 \text{ bits}$$

$$Y(CPS) = 9600/11$$

$$\text{Frame Width} = 11/Y(CPS) = .00114 \text{ s.}$$

The basic operation of the RS232C data link is controlled through the programmable ACIAs on both the AIM 65 interface board and the Apple RS232C interface board. The registers of the ACIA on the AIM 65 have been inserted in memory locations HEX 1000 through HEX 1008. The HEX address 1000 is the assigned base address and all the internal registers are accessed relative to this base address. The software developed to initiate the AIM 65 ACIA module first executes a reset operation, then writes the baud rate, word length, and the number of stop and start bits into the R6551 ACIA control register. Also, the internal clock rate is selected by loading a logic one into bit four of the ACIA control register.

A simple driver program is shown in Figure 5. Note that ASCII HEX 34 is sent to the transmit register on the ACIA via the STA 1000 command after first loading the value immediately into the accumulator using the LDA # 34 command.

```
K * = 400
/12
0400 A9 LDA #00
0402 8D STA 1001
0405 8D STA 1000
0408 A9 LDA #1E
040A 8D STA 1003
040D A9 LDA #05
040F 8D STA 1002
0412 A9 LDA #34
0414 8D STA 1000
0417 4C JMP 0412
041A 00 BRK
041B 00 BRK
```

Figure 5. ACIA driver program.

7

At this point, the data are sent to the Apple's ACIA and received in register HEX COAD for later use by the Main Matrix Program in the Apple microprocessor.

Now as to the data transfer between the state sensing AIM 65.1 and the state driving AIM 65.2, the difference in data transfer between the AIMs and the Apple is due to the use of the shift register on the AIM 65s. The angle change or relative position is sent out in a 96-bit data stream consisting of four blocks of 24 bits of information. Each block contains the 10-bit word that is a one-to-one correspondence with the absolute position and another 8 bits used for addressing and sign. The serial driver is shown using one memory location in Figure 6. Note that the 8-bit value is loaded from location :0400 to the shift register and is outputted on CB2 with CB1 containing the timing information of the shift pulses. There is also a dummy load to location :A00A for initialization of the shift register.

```
K * = 0FDC
/15
0FDC A9 LDA #00
0FDE 8D STA A008
0FEI A9 LDA #14
0FE3 8D STA A008
0FE6 A9 LDA #20
0FE8 8D STA A008
0FEB A9 LDA #00
0FED 8D STA A009
0FF0 AD LDA 0401
0FF3 8D STA A00A
0FF6 EA NOP
0FF7 EA NOP
0FF8 EA NOP
0FF9 00 BRK
0FFA 00 BRK
```

Figure 6. Serial driver program.

8

With the completion of the data transfer between the two AIM 65s, the feedback loop is completed, which is necessary for the system program to operate.

At this point, it becomes necessary to consider specifically the behavior of the serial driver and serial receiver programs under multiple 96-bit data transfer. The multiple data transfer is initiated by the addition of a multiple status flag check using the X and Y registers. The LDX #00 and LDY #00 commands each load a HEX value of zero into these registers which are subsequently incremented within the actual data transmission portion of the code and the multiple transfer portion of the code. It had become apparent after several data transmission efforts, that the inner loop or the delay time between each 8-bit word needed to be controlled severely such that data transmission would occur only after a handshake had been initiated and acknowledged after each 8-bit word rather than 12 words or 96 bits. With this software control in place, data drop outs were decreased but not totally eliminated. At present, the data loss is being handled by utilizing an error checking technique. The error checking technique consists of first determining if the encoder address is in correct sequence and then, once this check is passed proceeding to evaluate the magnitude of the current state change with the preceeding one in order to determine if the data is reasonable.

Transmission errors are a problem with medium-speed devices in which several methods have been developed to lessen the likelihood of such errors; error checking is one such method. Several others are:

a. Sampling the input data at the center of the transmission interval in order to avoid edge effects; that is, keeping away from the edges where that data is unstable.

b. Generating a checking parity; the use of an extra bit to force the correct data even or odd.

c. Longitudinal and Cyclic Redundancy checks.

At present, considerable effort is being made to minimize the transmission errors and provide a relatively stable control signal to drive the system.

V. STATE DRIVING [8]

Since the AIM 65.2 microcomputer controller of the electro-hydraulic valves has no position sensors connected directly to it, position data must be received from AIM 65.1 so that appropriate voltages can be sent to the power digital to analog converters which, in turn, control the electro-hydraulic valves. This AIM 65.1/65.2 interface consists of a four-line serial data port that is used to transmit serial data between each of the AIM 65s' 8-bit shift registers, where two lines from AIM 65.2 (out of four available) are hardwired to the corresponding two lines on the AIM 65.1. One line pair is used to transfer the control clock pulses while the other pair is used to transfer the serial data (encoder address and associated angular difference data). The handshake control of the data transfer is done using two lines (PB6 and PB7 on AIM 65.2 to PB4 and PB5 on AIM 65.1, respectively). The serial data consists of a 96-bit data stream with the encoder address sent in one 8-bit field and the associated angular difference data sent in two 8-bit fields (contiguous to the encoder address).

Note that each 8-bit field, as it is received, is stored in consecutive memory locations so that the AIM 65 control program can, when data is needed, sequentially pick out of memory the encoder address and angular difference data.

After receiving the serial data, the control program written in the BASIC programming language resident on the AIM 65, then "PEEKS" to the memory locations holding the 1st encoder address (eventually, the 2nd, 3rd, and 4th) and associated angular difference data and converts these binary values to decimal equivalent. The results are decoded and checked for errors. If no errors are found, the program proceeds to find the power digital-to-analog converter (PDAC) output voltage versus angular difference equation associated with each link.[9] From this equation, the PDAC voltage is found which corresponds to that angular difference condition. This value is then converted to a decimal number based on the resolution of the PDAC (11 effective bits at ± 24-V power transistor supply voltage or $24/(2^{11} - 1) = 11.7$ mV/bit change) and with the encoder/link address, outputted to the parallel output port of the AIM 65.2. This output port actually consists of two 8-bit ports, Port A (PA) and Port B (PB) as well as the four serial communications lines. The eight least significant bits of the voltage value are outputted on Port A, while the remaining three most significant bits of the voltage are outputted to Port B along with the sign bit and the 2 bits used for the encoder/link address. (The remaining 2 bits on Port B are used for handshake control of the serial data transfer.)

Following the output of address and data (including sign bit) to Ports A and B, is a strobe of several milliseconds duration that "enables" the address decoder which, in turn, sends a strobe to the addressed PDAC.

---

[8] The reason for off loading this function to another microcomputer was the resulting increase in control speed due to parallel processing.

[9] An empirical equation found by measuring the response of each link to an applied voltage.

Now, if errors are found in the decoder address and data, the program would shutdown the backhoe, print a message stating the reason for shutdown, and query the user as to what to do next. The program has two responses to the user's reply: terminate program or begin again by requesting more data from AIM 65.1.

This process is continued until the angular differences, given for each link, approach a tolerance value[10] close to 0, corresponding to 0 V on the output voltage-angular difference equation. If a link does overshoot the final position, the angular difference data received from AIM 65.1 will be decoded as a negative value which corresponds to a negative PDAC output voltage. Consequently, the direction of link motion will be reversed towards the correct final position. A block diagram of the output port of the AIM 65.2 and all that is connected to it are shown in Figures 7 and 8, respectively.

The power digital-to-analog converter consists of a 12-bit storage register with strobed inputs, a 12-bit digital-to-analog converter, and a power output stage (see Figure 9). The maximum drive voltage the servo valves will receive from the PDACs is 15 V at 100 percent connect time.[11] The resolution of the PDACs given a supply voltage of 24 V to the power stage and an effective word length of 11 bits, is 11.7 mV. Maximum drive current that the PDAC can continuously supply to a load (in this case, a 27-ohm solenoid) is ± 1.2 A.

The electro-hydraulic valve that replaced the case valve on the case backhoe was the Monsu-Tison HV03 proportionally controlled valve with an attached Monsun-Tison EHC3 electro-hydraulic converter; optional joysticks were added. With these valves, a nearly linear output hydraulic pressure input vs. drive current can be attained.

---

[10] An empirical value based on the response of each link.

[11] A higher voltage can be used (up to 24 V) if the voltage source is pulsed (50 percent duty cycle). This reduces the mechanical hysteresis that is developed in the solenoids of the EHC3. Also, with the higher voltage, an increase in backhoe arm "power" can be realized.

11
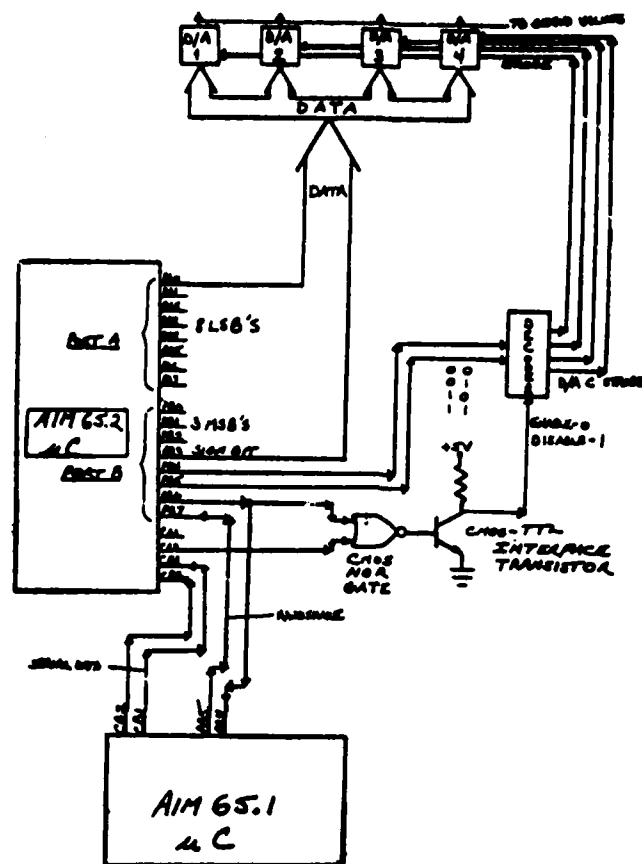
Figure 7. AIM 65.2 output port.
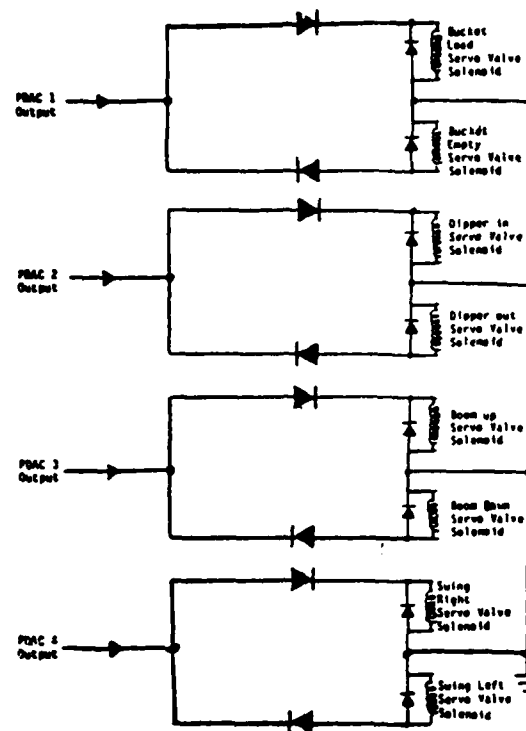
Figure 8. PDAC output-to-servovalve solenoid.

13

Figure 9. PDAC AND INPUT CODE table.
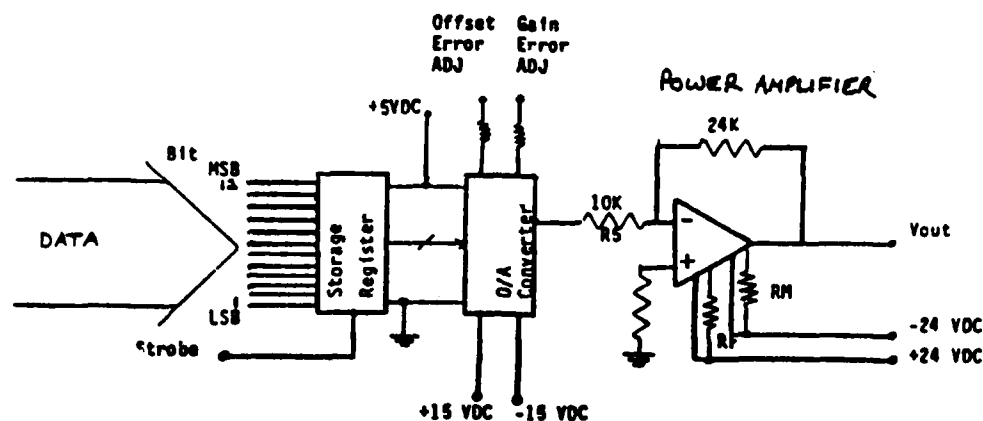
	Input Code (offset binary)		Output
SB	MSB's	LSB's	Voltage
1	111	11111111	+24.00V
1	111	11111110	+23.988V
1	000	00000000	+11.7 MV
0	111	11111111	0.000V
0	111	11111110	-11.7 MV
0	000	00000000	-23.976V
0	000	00000000	-23.988V

14

VI. HOLE STATE DETERMINATION

During the initial stages of work with the automated backhoe, the overall system controller will be "blind" as to the hole state. The backhoe will be digging in soft earth so the controller will, at this point, assume that the bucket is removing a cubic matrix square with each bucket-full of earth and, when finished with the complete digging cycle, assume the desired hole was dug to the specifications inputted by the user. But, after successful demonstration of the complete system controller, a sensor(s) will be added to supply this information to the controller. A sensor that was determined could supply this information (hole state and bucket digging efficiency) is a rangefinder.

Using a rangefinder, the controller would be capable of making a hole survey (if rangefinder was properly mounted) by comparing the three-dimensional hole matrix that it generated in its control program to the results of the rangefinder's scan. Also, the controller could determine how much material the bucket was removing (bucket digging efficiency) by curling the bucket to a predetermined position, then using a rangefinder (again, properly mounted) to indicate the level of material in the bucket by examining the distance between the sensor and the material in the bucket.

In evaluating the test results conducted on a specific rangefinder (Ultrasonic Ranging type), several problem areas occurred. The most significant of these was the 20-degree beam width of the ranger. This beam width would result in a detectable cone diameter of 4.21 ft at a distance of 12 ft. It is clear that this large beam width would result in too little resolution making a hole survey. Therefore, the Ultrasonic Ranging System tested was *not used during the control cycle* tests.

VII. RESPONSE OF SYSTEM

The results presented in this report are preliminary since the full Control System Modeling is expected to be presented in another report in the near future.

Using basic principles, most dynamic systems are subject to analysis using a force law formulation of the equations of motion. The force law can be stated simply as:

$$\frac{d^2}{dt^2} m_i \vec{r}_i = \vec{F}_i^{(e)} + \sum_j \vec{F}_{ji},$$

where: $\vec{F}_i^{(e)}$ represents external forces to the ith particle, and

$$\sum_j \vec{F}_{ij} \text{ represents internal forces with } \vec{F}_{jj} = 0.$$

If all the forces are summed over all particles then

$$\sum_i \frac{d^2}{dt^2} m_i \vec{r}_i = \sum_i \vec{F}_i^{(e)} + \sum_{ij} \vec{F}_{ji} \longrightarrow 0 \ \{ \ i \neq j, \text{ and}$$

$$\frac{d^2}{dt^2} \sum_i m_i \vec{r}_i = \sum_i \vec{F}_i^{(e)}.$$

Rewriting with center of mass considerations,

$$\vec{R} = \frac{\Sigma M_i \vec{r}_i}{\Sigma M_i} = \frac{\Sigma M_i \vec{r}_i}{M} \ ,$$

$$M\vec{R} = \Sigma M_i \vec{r}_i,$$

$$\frac{d^2}{dt^2} M\vec{R} = \sum_i \vec{F}_i^{(e)} \ ,$$

Therefore, $M \dfrac{d^2}{dt^2} \vec{R} = \sum_i \vec{F}_i^{(e)} = \vec{F}^{(e)}.$

Using a force law formulation, what is needed is an accurate mathematical representation of the system we intend to control. Clearly, there are several major elements; these are shown in Figure 10.
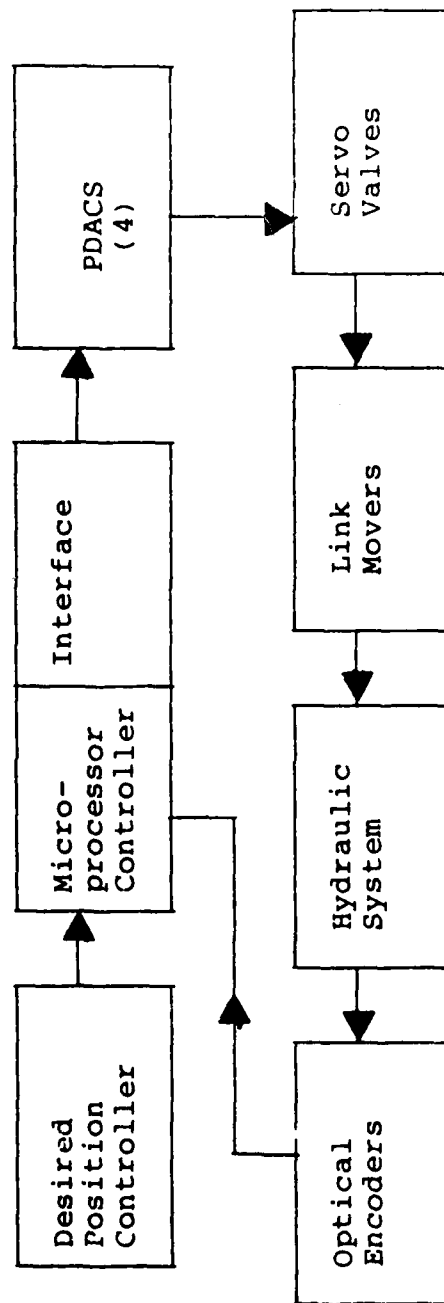
16

Figure 10. Major elements of controlled system.

17

Essentially, the control flow is as follows. The reference or desired position signal is entered into the microprocessor controller section. This section consists of the three 6502 CPUs, two AIM 65s, and one Apple microprocessor, with the necessary intercommunication hardware established as a separate interface within this section. The controller section produces the error signal or the difference, in terms of angles, between the actual position of the backhoe arm vs. the final desired position. As stated previously, intercommunication between the AIM 65 microprocessors is based on a 96-bit data stream of which 10 bits represent the error, and 12 bits are used to drive the electro-hydraulic servo valves. This error signal (12-bit) is decoded by each selected high-power digital-to-analog converter (PDAC) and the resulting analog voltage at the output of the selected PDAC is placed across the input of the electro-hydraulic servos. The hydraulic flow utilized by the servos then drives the main cylinder flow which is used to actuate the mechanical links of the backhoe. At this point, the machine is being driven to its final programmed position with disturbances being placed at this section of the Control System.

The return loop closure is provided by four optical encoders mounted at each link joint such that the final state of the arm can be calculated easily.

Now that the Control System block diagram has been discussed, a preliminary quantitative analysis as to the stability of the total System will be discussed with the emphasis of analysis being placed in the electro-hydraulic valve section.

The input or desired position signal is entered manually using the AIM 65.1 ASCII Keyboard, into eight consecutive memory locations. This information is used to calculate the difference between the final state and the initial state. The time delay for this multiprecision subtraction, as well as configuration of the data stream to include enabling the optical encoders sequentially, is approximately 700 Hz or 0.001 s. Currently, the actual program has an additional 0.030-s delay that is being introduced as a short term solution to the data loss during transmission of the calculated difference angles. Therefore, the error signal generated and sent to the input of each of the PDACs is subject to a time delay of approximately 0.060 s total (also includes the state driving portion of the delay loop). We can now replace the first two blocks of Figure 10 as shown in Figure 11.

Now the PDACs are Burr-Brown Model 4804, 12-bit power, digital-to-analog converters. Using the specifications provided by Burr-Brown, the maximum settling time for any input change is stated as 0.001 s with a maximum output resistance of 1 ohm. The next block in Figure 10 can be replaced by essentially a single time delay of 0.001 s as shown in Figure 12.
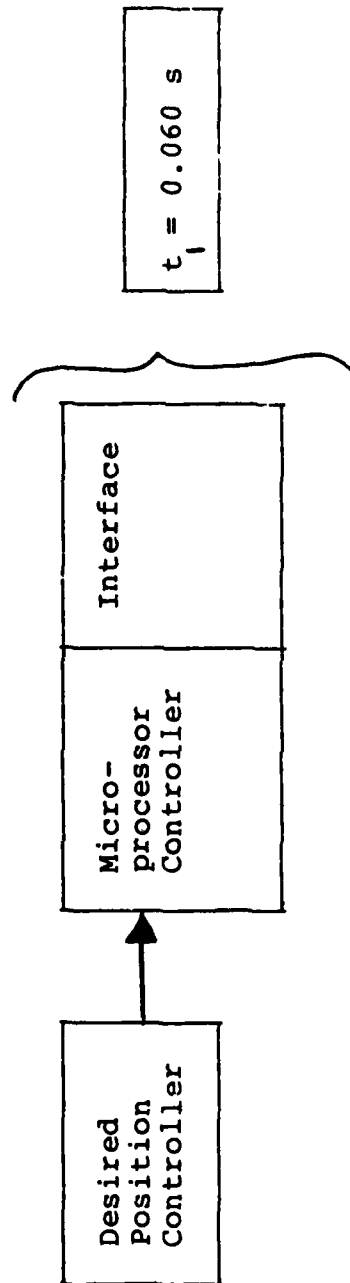
Figure 11. Timing t_1.

19

```
┌─────────────────────┐        ⎫              ┌─────────────────────┐
│                     │        ⎪              │   t₂                │
│      PDACS          │        ⎬              │   .001 s            │
│                     │        ⎪              │                     │
└─────────────────────┘        ⎭              └─────────────────────┘
```
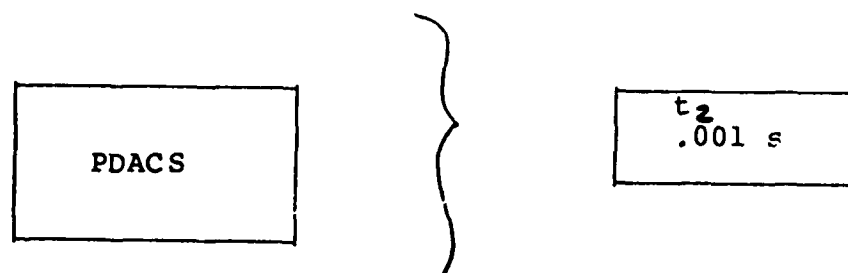
Figure 12. Timing t_2.

At this point, we have now reduced all the functional blocks prior to the electro-hydraulic servo valves to a series of time delays that act upon the control signal. The next step is to model the response of the electro-hydraulic valves which is clearly a third order system. Since the complete analysis for the electro-hydraulic valve has been treated in detail by many authors,[12] only the results of the transient response characteristics will be presented here.

The essential operation of the electro-hydraulic servo valves is as follows. If mechanical motion is desired in one direction or the other, a current is applied to the windings of a coil in which the armature is rigidly attached to the spool of the valve. This spool acts to cover or uncover (depending on the direction) one of three ports of the pilot hydraulic pressure loop. When the appropriate port is uncovered, hydraulic fluid is conducted at 5 gal/min from the supply to one of the chambers of the ram cylinder, thus, moving the mechanical links. (See Figure 13.)

---

[12] R. N. Clark, "Introduction to Automatic Control Systems," Wiley, 1962.

20
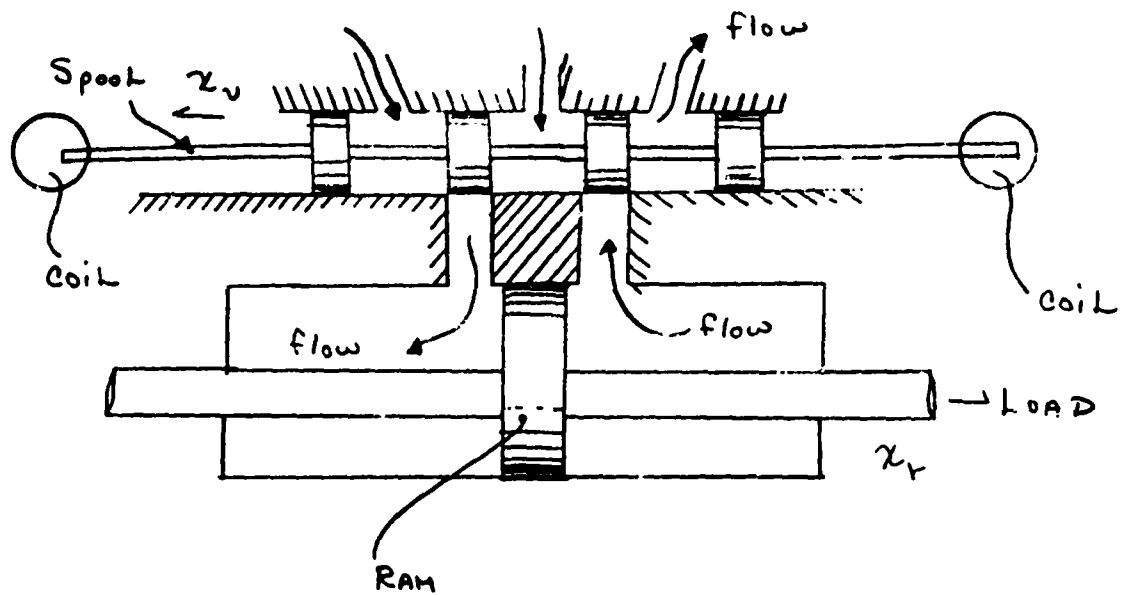
Figure 13. Electro-hydraulic valve.

Utilizing an equivalent circuit representation of this physical system and operating the coils in a push-pull fashion, the difference in current relationship results in the following transfer function:

$$\frac{I_d(s)}{E_i(s)} = \frac{2\mu L}{s + (r_{PDAC} + R_L)/L}$$

where: μ = amplification factor,
$\quad$ L = inductance of the coil,
$\quad$ I_d = differential current,
$\quad$ E_i = applied current,
$\quad$ R_L = resistance of windings, and
$\quad$ r_{PDAC} = resistance of PDACs.

21

With this push-pull operation, the force developed on the spool is defined in relation to the differential current dI:

$$F = K_M I_d.$$

where: F = net force on spool,

K_M = magnetic force constant, and

I_d = differential current.

This force developed on the spool will result in motion of the spool according to the following equation:

$$F = M \frac{d^2}{dt^2} \chi_\nu + \beta \frac{d\chi_\nu}{dt} + K\chi_\nu + F_R$$

where: M = mass of the spool armature assembly,

β = equivalent viscous friction,

K = restoring spring coefficient, and

F_R = friction force on spring.

Deriving the transfer function relating spool displacement to applied force results in:

$$\frac{X_\nu(s)}{F(s)} = \frac{1/M}{s^2 + (B/M)s + K/M}$$

Combining the physical parameters flow, ram velocity and wetted area of the piston face results in:

$$\dot{\chi}_r(t) = EK \left\{ 1 - \frac{1}{\alpha^2 - 2\xi W_n \alpha + W_n^2} e^{-\alpha t} + \frac{\alpha e^{-\xi W_n t} \sin(W_o t + \psi)}{(1-\xi^2)^{1/2}(\alpha^2 - 2\xi\alpha W_n + W_n^2)^{1/2}} \right\}$$

$$:\chi = -\tan^{-1}\left(\frac{(1-\xi^2)^{1/2}}{-\xi}\right) -\tan^{-1}\left(\frac{W_o}{\alpha - \xi W_n}\right) \quad :\alpha = \frac{R(ohms)}{L(henry)}$$

$$:W_o = W_n(1-\xi^2)^{1/2} \quad :W_n^2 = \frac{K}{M} \quad :\xi = \frac{B}{2(KM)^{1/2}}$$

22

What can be stated at this point is that there appears to be no significant contribution to the control loop structure from the electro-hydraulic valves for times greater than .14 s. Therefore, the servo valves section is now clearly defined by the equation for ram velocity.

The next two blocks consisting of the link-movers and hydraulic system are machine dependent and the subject of considerable modeling. Therefore, this section will be presented in detail during October of FY84.

Using the program in Appendix C, the ram velocity $\chi_r(t)$ is plotted as a function of time with the result shown in Figure 14. The significance is that the solution of this third order system has resulted in a fairly well damped curve with no overshoot of the steady state valve.

Including the optical encoder section, essentially a time delay only, the control loop structure has been currently determined to be as shown in Figure 15.

VIII. EXPERIMENTAL RESULTS

This section is included for completeness only since evaluation of the system is still in progress. The preliminary results from several software sequence tests indicate that data dropouts and the necessary error checking are adding an additional 2 to 4 s to the time needed for the state driving program to update its driving voltages. Therefore, the desired final position is exceeded. Several alternatives have been proposed to the use of the 6522 VIA shift registers. Currently, a switching from serial to parallel intercommunication between the two AIM 65s is in progress.

IX. CONCLUSIONS

The principle findings of this review are as follows:

Currently the state determinations, state sensing, and data transfer (96-bit stream) programs have been designed, developed, and tested on the prototype demonstrator. All hardware modifications have been completed.

In the future, it is expected that a complete open loop/closed loop servo response with a full system will be analyzed to determine deviations from expected valve due to the mechanical uncertainities in the system and the maximum drive rate of the hydraulic system.

23

Ram Velocity
(cm/sec) 81o288

TIME(sec).88811o.14

RESISTANCE(OHM).   50

MAGNETIC FIELD(DYNE/CM/SEC).   600
SPRING
COEFFICIENT(DYNE/CM).   133088
MASS(GRAM).   34
INDUCTANCE(HENRIES).   5
AMP FACTOR(VOLT/VOLT).   200

ALPHA.   16.8666666687
FREQ.   62.5441828871
FREQ(8).   61.0185758444
PSI.   -0.8185515700288
COEFFICIENT 1.   1.98419463887
COEFFICIENT 2.   0.269734649364

MAX VALUE X(T)
VELOCITY (CM/SEC). 191.187725364

Figure 14. Transient response.

24
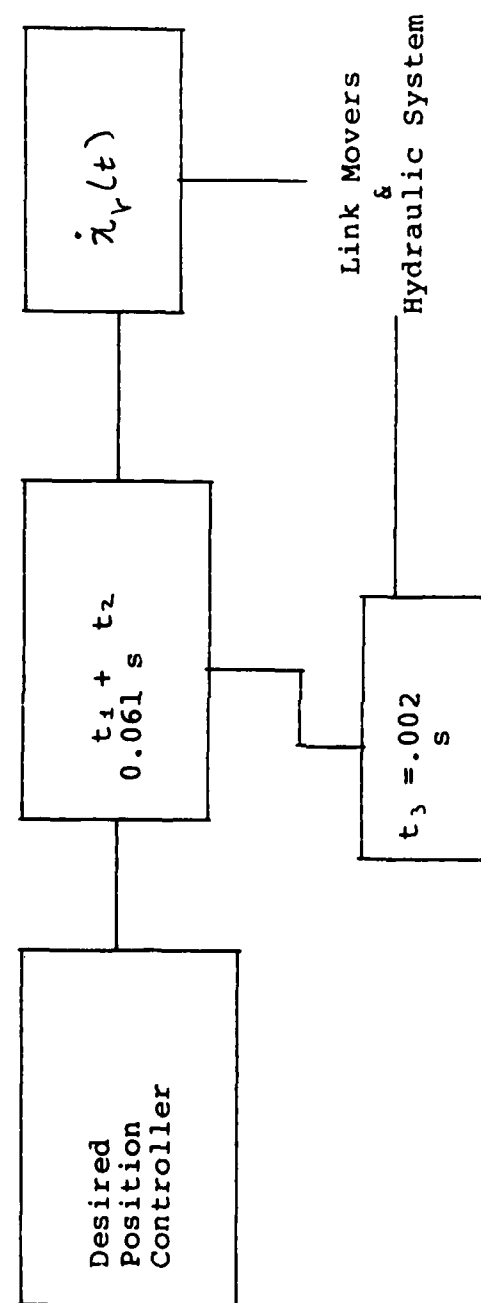
Figure 15. Control structure/timing.

Additionally, it is anticipated that considerable work needs to be done in interfacing the main control program and controller so that considerations such as:

a. Reference point motion,

b. Inertial load changes,

c. Real time response, and

d. Multiple processing

are included in the prototype demonstration.

APPENDIX A

STATE DETERMINATION

```
PROGRAM;
STATE DETERMINATION;

(K)*=200
/99
0200 A9 LDA #00
0202 8D STA A003
0205 A9 LDA #2C
0207 8D STA A002
020A A9 LDA #00
020C 8D STA A000
020F AD LDA A001
0212 8D STA 0400
0215 AD LDA A000
0218 8D STA 0401
021B A9 LDA #04
021D 8D STA A000
0223 8D STA 0402
0226 AD LDA A000
0229 8D STA 0403
022C A9 LDA #08
022E 8D STA A000
0231 AD LDA A001
0234 8D STA 0404
0237 AD LDA A000
023A 8D STA 0405
023D AD LDA #0C
023F 8D STA A000
0242 AD LDA A001
0245 8D STA 0406
0248 AD LDA A000
024B 8D STA 0407
024E EA NOP
```

```
┌─────────────┐
│  Set up     │
│             │
│  Ports VIA  │
└─────────────┘
       │
       ▼
┌─────────────┐
│  Configure  │
│             │
│  Pinouts    │
└─────────────┘
       │
       ▼
┌─────────────┐
│  Enable     │
│             │
│  Encoders   │
└─────────────┘
       │
       ▼
┌─────────────┐
│  Store      │
│             │
│  10bit word │
└─────────────┘
```
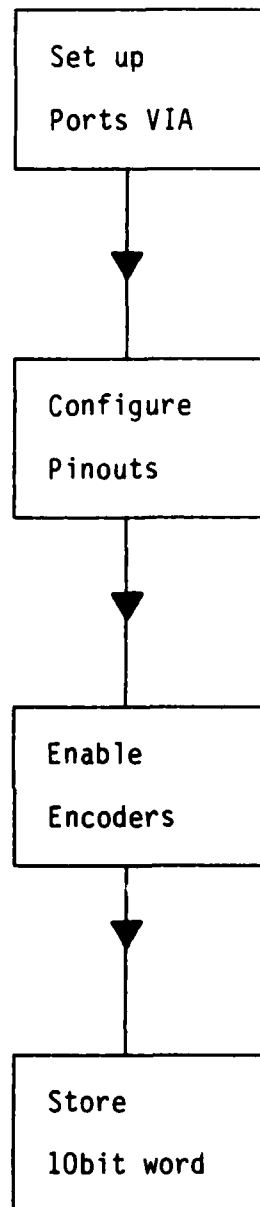
Encoder Select/Data Read Program
Completed Jan 82, R.J. Chavez

```
024F 38 SEC
0250 AD LDA 0408
0253 ED SBC 0400
0256 8D STA 0410
0259 AD LDA 0409
025C ED SBC 0401
025F 8D STA 0411
0262 90 BCC 02C0
0264 38 SEC
0265 AD LDA 040A
0268 ED SBC 0402
026B 8D STA 0412
026E AD LDA 040B
0271 ED SBC 0403
0274 8D STA 0413
0277 90 BCC 02CF
0279 38 SEC
027A AD LDA 040C
027D ED SBC 0404
0280 8D STA 0414
0283 AD LDA 040D
0286 ED SBC 0405
0289 8D STA 0415
028C 90 BCC 02DF
028E 38 SEC
028F AD LDA 040E
0292 ED SBC 0406
0295 8D STA 0416
0298 AD LDA 040F
029B ED SBC 0407
029E 8D STA 0417
02A1 90 BCC 02EF
02A3 EA NOP
02A4 EA NOP
02A5 4C JMP 0300
02A8 EA NOP
02A9 EA NOP
02AA EA NOP
02AB EA NOP
```

Set up

final 10 bits

Multiprecision
subtraction
(16 bits)

Store
Results

(+)

No change
ADJ. Sign

(-)

Change ADJ
Sign
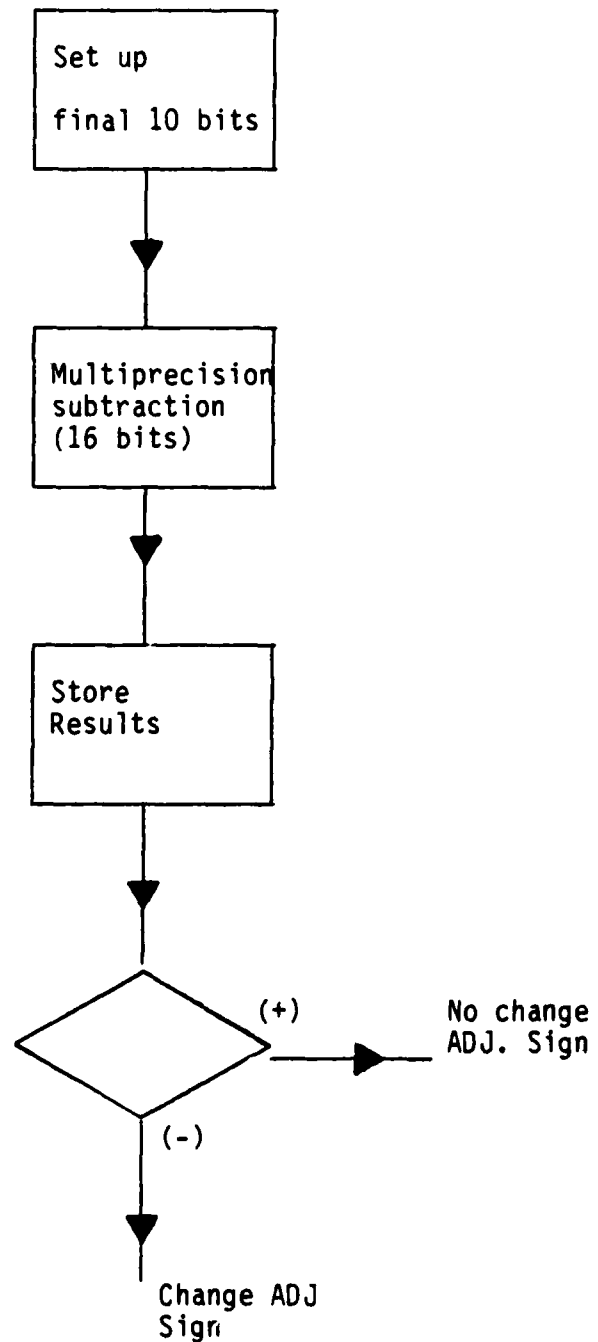
Multiprecision Routine for State Data
Completed Oct 82.   R.J. Chavez

.

29

```
02AC EA NOP                                   02D7 4C JMP 0279
02AD EA NOP                                   02DA EA NOP
02AE EA NOP                                   02DB EA NOP
02AF EA NOP                                   02DC EA NOP
02B0 EA NOP                                   02DD EA NOP
02B1 EA NOP                                   02DE EA NOP
02B2 EA NOP                                   02DF AD LDA 0415
02B3 EA NOP          ┌──────────────┐         02E2 09 ORA #04
02B4 EA NOP          │ Sign Conf.   │         02E4 8LD STA 0415
02B5 EA NOP          │ ADJ.         │         02E7 4C JMP 028E
02B6 EA NOP          │ Bits         │         02EA EA NOP
02B7 EA NOP          │              │         02EB EA NOP
02B8 EA NOP          └──────────────┘         02EC EA NOP
02B9 EA NOP                  │                02ED EA NOP
02BA EA NOP                  │                02EE EA NOP
02BB EA NOP                  ▼                02EF AD LDA 0417
02BC EA NOP                                   02F2 09 ORA #04
02BD EA NOP          ┌──────────────┐         02F4 8D STA 0417
02BE EA NOP          │              │         02F7 4C JMP 0300
02BF EA NOP          │  Continue    │         02FA EA NOP
02C0 AD LDA 0411     │              │         02FB EA NOP
02C3 09 ORA #04      │              │         02FC EA NOP
02C5 8D STA 0411     └──────────────┘         02FD EA NOP
02C8 4C JMP 0264                              02FE EA NOP
02CB EA NOP                                   02FF EA NOP
02CC EA NOP
02CD EA NOP
02CE EA NOP
02CF AD LDA 0413
02D2 09 ORA #04
02D4 8D STA 0413
```

Expansion Area & Sign Manipulation
Completed Dec 82, R.J. Chavez

30

```
0300 A2 LDX #00                              C373 EA NOP
0302 0E ASL 0411                             0374 EA NOP
0305 0E ASL 0413                             0376 EA NOP
0308 0E ASL 0415                             0377 EA NOP
030B 0E ASL 0417                             0378 EA NOP
030E E8 INX                                  0379 EA NOP
030F E0 CPX #05                              037A EA NOP
0311 F0 BEQ 0316                             037B EA NOP
0313 4C JMP 0302                             037C EA NOP
0316 A2 LDX #00                              037D EA NOP
0318 4E LSR 0411                             037E EA NOP
031B 4E LSR 0413                             037F EA NOP
031E 4E LSR 0415                             0380 EA NOP
0321 4E LSR 0417                             0381 EA NOP
0324 E8 INX                                  0382 EA NOP
0325 E0 CPX #05                              0383 EA NOP
0327 F0 BEQ 032C                             0384 EA NOP
0329 4C JMP 0318                             0385 EA NOP
032C EA NOP                                  0386 EA NOP
032D EA NOP                                  0387 EA NOP
032E EA NOP                                  0388 EA NOP
032F A9 LDA #00                              0389 EA NOP
0331 8D STA 0600                             038A EA NOP
0334 AD LDA 0410                             038B EA NOP
0337 8D STA 0601                             038C EA NOP
033A AD LDA 0411                             038D EA NOP
033D 8D STA 0602                             038E EA NOP
0340 A9 LDA #10                              038F EA NOP
0342 8D STA 0603
0345 AD LDA 0412
0348 8D STA 0604
034B AD LDA 0413
034E 8D STA 0605
0351 A9 LDA #20
0353 8D STA 0606
0356 AD LDA 0414
0359 8D STA 0607
035C AD LDA 0415
035F 8D STA 0608
0362 A9 LDA #30
0364 8D STA 0609
0367 AD LDA 0416
036A 8D STA 060A
036D AD LDA 0417
0370 8D STA 060B
```

```
+------------------+
| Set up           |
| MSB LSB          |
| Data Stream      |
|                  |
|                  |
+------------------+
         |
         v
+------------------+
| Set up           |
| Full Date        |
| St.              |
|                  |
+------------------+
```
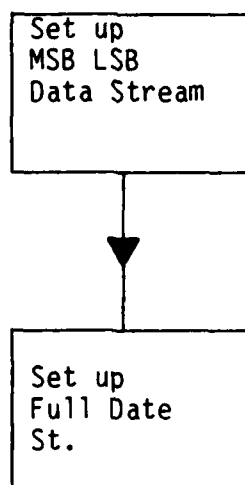
Expansion Area & 96 Bit (full) Data Stream Manipulation
Completed (1) Mar 82, Dec 82, R.J. Chavez

```
0390 18 CLC
0391 AD LDA A000
0394 4A LSR  A
0395 4A LSR  A
0396 4A LSR  A
0397 4A LSR  A
0398 0A ASL  A
0399 0A ASL  A
039A 0A ASL  A
039B 0A ASL  A
039C 0A ASL  A
039D 0A ASL  A
039E 0A ASL  A
039F 4A LSR  A
03A0 4A LSR  A
03A1 4A LSR  A
03A2 EA NOP
03A3 EA NOP
03A4 EA NOP
03A5 EA NOP
03A6 EA NOP
03A7 EA NOP
03A8 EA NOP
03A9 C9 CMP #10
03AB F0 BEQ 03B0
03AD 4C JMP 0200
03B0 A9 LDA #20
03B2 8D STA A000
03B5 20 JSR ED2C
03B8 20 JSR ED2C
03BB 20 JSR ED2C
03BE 20 JSR ED2C
03C1 20 JSR ED2C
03C4 20 JSR ED2C
03C7 EA NOP
03C8 EA NOP
03C9 EA NOP
03CA EA NOP
03CB EA NOP
03CC EA NOP
03CD EA NOP
03CE EA NOP
03CF EA NOP
```

Load PB4
Rotate bit
for (+) ID

ASL A

LSR A

PB4
on   N

Y

go to transmit routine
SET Bit HI
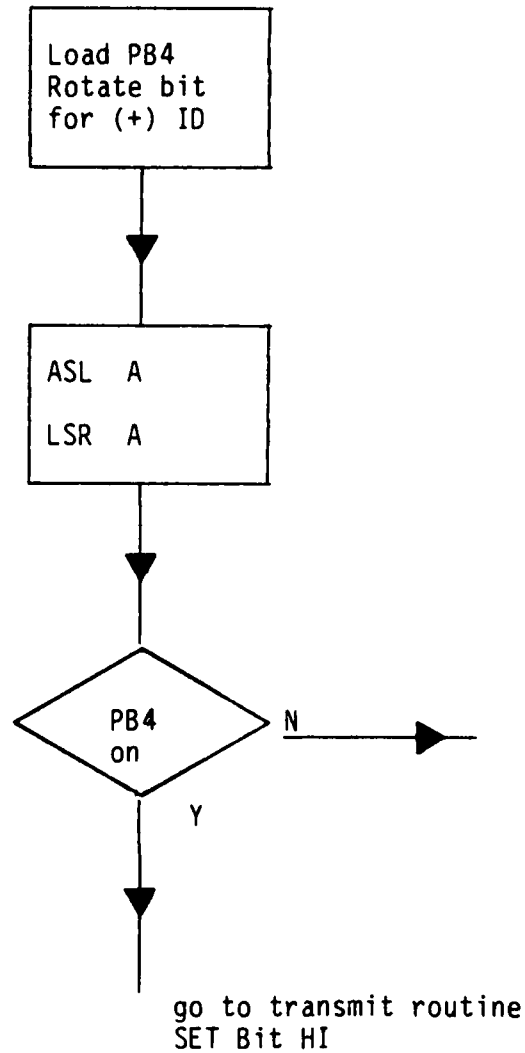
Data Transfer Handshake Routine Completed
Single: April 82, Multi: (a) Dec 82, (b)
R.J. Chavez

```
03D0 A2 LDX #00
03D2 A0 LDY #00
03D4 A9 LDA #00
03D6 8D STA A00B
03D9 A9 LDA #14
03DB 8D STA A00B
03DE A9 LDA #20
03E0 8D STA A008
03E3 A9 LDA #00
03E5 8D STA A009
03E8 BD LDA 0600,X
03EB 8D STA A00A
03EE C8 INY
03EF C0 CPY #FF
03F1 D0 BNE 03EE
03F3 E8 INX
03F4 E0 CPX #0C
03F6 D0 BNE 03D2
03F8 A9 LDA #00
03FA 8D STA A000
03FD 00 BRK
```


Transmit
96 bit
stream
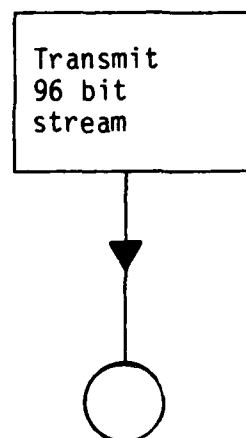
Data Transfer Routine Completed
Single: (a) April 82, Multi: (a) Dec 82 (15), (b)
R.J. Chavez

33

APPENDIX B

PROGRAM CONTROL

```
PROGRAM CONTROL;
 USES TRANSCEND;

CONST A2MAX=75;A4MIN=0;MHL=16;MHW=16;MHD=6;(*MAX HOLE DIM'S*)
 BOLE=8.4;DILE=6.1;BULE=3.2;BUWI=1.5;BUDE=1.5;
TYPE SPEC=ARRAY[1..MHL,1..MHW] OF INTEGER;

VAR CM,DIMODE,DUMODE:CHAR;
    HOSPEC,SPSPEC,HOSTATE,SPSTATE:SPEC;
    A1,A2,A3,A4,AI1,AI2,AI3,AI4,
    D1,D2,D3,D4,I,J,K,L,W,D,POT1,POT2,POT3,POT4,:INTEGER;
    BL,BW,BD,X,Y,Z:REAL;

SEGMENT PROCEDURE HOLESPEC;
 (*CREATES SPEC*)
 VAR P:INTEGER;
 BEGIN
  WRITELN('INPUT HOLE LENGTH, INTEGER<',MHL);
  READLN(L);
  P:=1+TRUNC(L/BULE);
  BL:=L/P;
  L:=P;
  WRITENLN('INPUT HOLE WIDTH, EVEN INTEGER<',MHW);
  READLN(W);
  P:=1+TRUNC(W/BUWI);
  BW:=W/P;
  W:=P;
  WRITELN('INPUT HOLE DEPTH, INTEGER<',MHD;
  READLN(D);
  P:=1+TRUNC(D/BUDE);
  BD:=D/P;
  D:=P;
  FOR I:=1 TO MHL DO
  FOR J:=1 TO MHW DO
   IF (J>(MHW/2-W/2)) AND (J<(MHW/2+W/2)) AND (I<L)
    AND(SQRT(SQR(I*BL)+SQR(J*BW)+SQR(K*BD))<BOLE+DILE+BULE)
    THEN HOSPEC[I,J]:=D
   ELSE HOSPEC[I,J]:=0;
 END;

SEGMENT PROCEDURE SPOILSPEC;
 (*CREATES SPEC*)
 VAR C:CHAR;
     P,L,W:INTEGER;
 BEGIN
  WRITELN('SPOIL TO R/IGHT, L/EFT OR B/OTH SIDES?');
  READ(C);
  WRITELN('INPUT LENGTH, INTEGER<',MHL);
  READLN(L);
  P:=1+TRUNC(L/BULE);
  BL:L/P;
  L:=P;
  WRITELN('INPUT WIDTH, INTEGER<',MHW);
```

35

```
        READLN(W);
        P:=1+TRUNC(W/BUWI);
        BW:=W/P;
        W:=P;
        FOR I:=1 TO MHL DO
         FOR J:=1 TO MHW DO
         IF I<L THEN
         CASE C OF
         'R':IF (HOSPEC[I,J]=0) AND (J>MHW/2) AND (J<(MHW-W)/2) AND
           (SQRT(SQR(I*BL)+SQR(J*BW)+SQR(K*BD))<BOLE+DILE+BULE)
           THEN SPSPEC[I,J]:=1  ELSE SPSPEC[I,J]:=0;

         'L':IF (HOSPEC[I,J]=0) AND (J<MHW/2) AND (J>(MHW-W)/2) AND
           (SQRT(SQR(I*BL)+SQR(J*BW)+SQR(K*BD))<BOLE+DILE+BULE)
           THEN SPSPEC[I,J]:=1  ELSE SPSPEC[I,J]:=0;

         'B':IF HOSPEC[I,J]=0
           (SQRT(SQR(I*BL)+SQR(J*BW)+SQR(K*BD))<BOLE+DILE+BULE)
           THEN SPSPEC[I,J]:=0;
         END
         ELSE SPSPEC[I,J]:=0;
        END;

    PROCEDURE QUIT;FORWARD;

    PROCEDURE ATDMP;FORWARD;

    PROCEDURE ATOD;EXTERNAL;

    PROCEDURE GETPOS;EXTERNAL;

    PROCEDURE SENDPOS;EXTERNAL;

    SEGMENT PROCEDURE MANUAL;
     (*PROVIDES JOYSTICK CONTROL*)
     CONST K=1;
     VAR C:CHAR;

      PROCEDURE TEMPLATE;
      BEGIN
        WRITELN('PROCEDURE TEMPLATE');
       IF (Z<0) AND ((X>L) OR (Y>W/2) OR (Y<-W/2) OR (Z<D)) THEN
       BEGIN
       A1:=AI1;
       A2:=AI2;
       A3:=AI3;
       A4:=AI4;
       END;
      END;
```

36

```
PROCEDURE LOCATE;
 VAR RP:REAL;
 BEGIN
  Z:=BOLE*SIN(A2)+DILE*SIN(A2+A3)+BULE*SIN(A2+A3+A4);
  RP:=BOLE*COS(A2)+DILE*COS(A2+A3)+BULE*COS(A2+A3+A4);
  X:=RP*COS(A1);
  Y:=RP*SIN(A1);
 END;

BEGIN
 REPEAT
 BEGIN
  GETPOS;
  ATOD;
  A1:=AI1+K*POT1;
  A2:=AI2+K*POT2;
  A3:=AI3+K*POT3;
  A4:=AI4+K*POT4;
  IF DIMODE='A' THEN
  BEGIN
   LOCATE;
   TEMPLATE;
  END;
  SENDPOS;
  (*IF DUMODE='A' THEN CHECKCURL;
  IF CURLED THEN ATDUMP;*)
 END;
 UNTIL C='Q';
END;

PROCEDURE DIG;
 BEGIN
 END;

PROCEDURE WAIT;
 BEGIN
  REPEAT
  GETPOS;
  UNTIL (AI1=A1) AND (AI2=A2) AND (AI3=A3);
 END;

PROCEDURE SWING;
 BEGIN
  A1:=TRUNC(ATAN(X/Y));
  WAIT;
  SENDPOS;
 END;
```

37

```
PROCEDURE REACH;
 VAR S,VEAN,VELE:REAL;
 BEGIN
  VELE:=SQRT(SQR(Y)+SQR(Z));
  VEAN:=ATAN(Z/Y);
  S:=(BOLE+DILE+VELE)/2;
  A2:=TRUNC(2*ATAN(SQRT((S-VELE)*(S-BOLE)/S*(S-DILE)))-VEAN);
  A3:=TRUNC(2*ATAN(SQRT((S-BOLE)*(S-DILE)/S*(S-VELE))));
  WAIT;
  SENDPOS;
 END;

PROCEDURE QUIT;
 BEGIN
  WRITELN('PROCEDURE QUIT');
  ATDMP;
  (*CENTER BOOM
  HOME
  PROMPT OPERATOR TO SECURE*)
 END;

PROCEDURE SWEEP;
 BEGIN
 END;

PROCEDURE ATDMP;
 BEGIN
  WRITELN('P ATDMP');
  FOR I:=MHL DOWN TO 1 DO
  BEGIN
   IF SPSTATE[I,MHW]-SPSTATE[I+1,1]=1 THEN
   BEGIN
    DUMP;
    SPSTATE[I,J]:=SPSTATE[I,J]+1;
   END;

PROCEDURE AUTO;
 VAR S:REAL;
 BEGIN
  WRITELN('PROCEDURE AUTO');
  FOR K:=1 TO D DO
  BEGIN
   FOR I:=1 TO MHL DO
   FOR J:=1 TO MHW DO
   BEGIN
    IF HOSPEC[I,J]<>0 THEN
    BEGIN
     A2:=A2MAX;
     A3:=90-A2;
     A4:=A4MIN;
     SENDPOS;
```

38

```
            Z:=BD*(1-HOSTATE[I,J]);
            Y:=BL*J;
            X:=BW*(I-8);
            SWING;
            REACH;
            DIG;
            HOSTATE[I,J]:=HOSTATE[I,J]+1;
            ATDMP;
          END;
        END;
       SWEEP;
      END;
    END;

   BEGIN (*MAIN*)
    PAGE(OUTPUT);
    WRITELN('E/XCAVATE OR Q/UIT?');
    READ(CM);
    IF CM='Q' THEN QUIT ELSE
    BEGIN
     WRITELN('M/ANUAL,T/EMPLATE, OR A/UTO MODE?');
     READ(CM);
     CASE CM OF
     'M':BEGIN
           DIMODE:='M';
           DUMODE:='M';
           MANUAL;
         END;

     'T': BEGIN
           WRITELN('M/ANUAL OR S/PECIFIED SPOIL DEPOSITION?');
           READ(CM);
           IF CM='S' THEN
           BEGIN
            DUMODE:='A';
            HOLESPEC;
            SPOILSPEC;
           END
           ELSE DUMODE:='M';
           MANUAL;
         END;

     'A': BEGIN
           HOLESPEC;
           SPOILSPEC;
           AUTO;
         END;

    END;
   END;
  QUIT;
  END.
```

```
                .PROC SENDPOS
                .PUBLIC A1,A2,A3,A4
RETURN          .EQU        0
                PLA
                STA         RETURN
                PLA
                STA         RETURN+1
                LDA         #03         ;RESET ACIA
                STA         OCOAO
                LDA         11          ;COM REG.
                STA         OCOAO
                LDA         A1
                PHA
                JSR         OUTPUT
                LDA         A1+1
                PHA
                JSR         OUTPUT
                LDA         A2
                PHA
                JSR         OUTPUT
                LDA         A2+1
                PHA
                JSR         OUTPUT
                LDA         A3
                PHA
                JSR         OUTPUT
                LDA         A3+1
                PHA
                JSR         OUTPUT
                LDA         A4
                PHA
                JSR         OUTPUT
                LDA         A4+1
                PHA
                JSR         OUTPUT
                LDA         RETURN+1
                PHA
                LDA         RETURN
                PHA
                RTS
OUTPUT          LDS         OCOAO
                AND         #2
                BNE         OUTPUT
                PLA
                STA         OCOA1
                RTS

                .END
```

```
            .PROC GETPOS
            .PUBLIC A1,A2,A3,A4
RETURN      .EQU    0
            PLA
            STA     RETURN
            PLA
            STA     RETURN+1
            LDA     #03     ;RESET ACIA
            STA     OCOAO
            LDA     11
            STA     OCOAO
            JSR     INPUT
            STA     A1
            JSR     INPUT
            STA     A2
            JSR     INPUT
            STA     A3
            JSR     INPUT
            STA     A4
            LDA     RETURN+1
            PHA
            LDA     RETURN
            PHA
            RTS

INPUT       LDA     OCOAO   ;GET STATUS
            LSR     A
            BCC     INPUT   ;LOOP TIL INPUT
            LDA     OCOA1   ;GET BYTE
            RTS
            .END
```

41

```
                    .MACRO POP
                    PLA
                    STA %1
                    PLA
                    STA %1+1
                    .ENDM

                    .MACRO WAIT
                    NOP
                    NOP
                    NOP
                    NOP
                    NOP
                    NOP
                    .ENDM

                    .PROC ATOD
                    .PUBLIC POT1,POT2,POT3,POT4
RETURN              .EQU        0
                    POP    RETURN
                    LDA    OCOCO
                    WAIT
                    LDA    OCOCO
                    STA    POT1
                    LDA    RETURN+1
                    PHA
                    LDA    RETURN
                    PHA
                    RTS


                    .END
```

APPENDIX C

PROGRAM FOR PLOTTING RAM VELOCITY

```
LIST
68 REM**********************************
70 REM****PROGRAM VALVE RESPONSE******
72 REM***** 6MAY83              ******
74 REM
76 REM***** BY: R. CHAVEZ        ****
78 REM*****      C. CLARK        ****
80 REM*                              *
82 REM**********************************
100 INIT
110 PAGE
120 A$="RESISTANCE(OHM)=                "
130 B$="MAGNETIC FIELD(DYNE/CM/SEC)="
140 C$="SPRING"
150 V$="COEFFICIENT(DYNE/CM)="
160 D$="MASS(GRAM)=                     "
170 E$="INDUCTANCE(HENRIES)="
180 F$="AMP FACTOR(VOLT/VOLT)="
190 PRINT A$
200 INPUT A1
210 PRINT B$
220 INPUT B1
230 PRINT C$
240 PRINT V$
250 INPUT C1
260 PRINT D$
270 INPUT D1
280 PRINT E$
290 INPUT E1
300 PRINT F$
310 INPUT F1
320 PRINT
330 PRINT A$,A1
340 PRINT B$,B1
350 PRINT C$
360 PRINT V$,C1
370 PRINT D$,D1
380 PRINT E$,E1
```

44

```
390 PRINT F$,F1
400 P$="ENTRY VALID =1 NOT=2"
410 PRINT
420 PRINT P$
430 INPUT P1
440 IF P1<1 THEN 100
450 IF P1>2 THEN 100
460 IF P1=2 THEN 100
470 REM NOTE THAT PROGRAM REQUIRES RADIANS
480 DIM X(1000),A(1000)
490 REMVIEWPORT 50,130,50,100
500 A2=A1/E1
510 C2=C1/D1
520 C3=C2↑0 5
530 B2=B1/(2*(C1*D1)↑0.5)
540 B3=B2↑2
550 B4=(1-B3)↑0.5
560 C4=C3*B4
570 R1=B4/(B2*-1)
580 R2=C4/(A2-B2*C3)
590 R3=-1*ATN(R1)-ATN(R2)
600 R4=C2/(A2↑2-2*B2*C3*A2+C2)
610 R5=A2/(B4*(A2↑2-2*B2*C3*A2+C2)↑0.5)
620 MOVE 40,45
630 SCALE 140/60,280/45
640 PAGE
650 AXIS 5,10
660 FOR I=1 TO 140
670 V=I/1000
680 X(I)=200*(1-R4*EXP(-A2*V)+R5*EXP(-B2*C3*V)*SIN(C4*V-R3))
690 REMA(I)=200*(1-EXP(-120*V)+0.41*EXP(-60*V)*SIN(294*V-3.14))
700 REM 2.49=143/57.3 AND 3.14=180/57.3
710 NEXT I
720 MOVE 1,X(1)
730 FOR I=1 TO 140
740 DRAW I,X(I)
750 NEXT I
760 MOVE 0,0
```

```
770 REMFOR I=1 TO 140
780 REMDRAW I,A(I)
790 REMNEXT I
800 HOME
810 MOVE 50,-40
820 S$="TIME(sec) .0001to.14"
830 PRINT S$
840 HOME
850 MOVE -50,180
860 T$="Rom Velocity"
870 U$="(cm/sec) 0to280"
880 PRINT T$
890 HOME
900 MOVE -50,160
910 PRINT U$
920 REMCOPY
930 REMPAGE
940 HOME
942 MOVE 20,-60
944 PRINT "*********************************************************"
950 MOVE 20,-70
960 PRINT A$,A1
970 MOVE 20,-90
980 PRINT B$,B1
990 MOVE 20,-100
1000 PRINT C$
1010 MOVE 20,-110
1020 PRINT V$,C1
1030 MOVE 20,-120
1040 PRINT D$,D1
1050 MOVE 20,-130
1060 PRINT E$,E1
1070 MOVE 20,-140
1080 PRINT F$,F1
1090 MOVE 20,-150
1100 PRINT
1110 MOVE 20,-160
1120 PRINT "ALPHA= ",A2
```

46

```
1130 MOVE 20,-170
1140 PRINT "FREO=",C3
1150 MOVE 20,-180
1160 PRINT "FREO(0)=",C4
1170 MOVE 20,-190
1180 PRINT "PSI=",R3
1190 MOVE 20,-200
1200 PRINT "COEFFICIENT 1=",R4
1210 MOVE 20,-210
1220 PRINT "COEFFICIENT 2=",R5
1230 MOVE 20,-220
1240 PRINT "**********************************************************
1250 MOVE 20,-240
1260 PRINT "MAX VALUE X(T)"
1270 MOVE 20,-250
1280 PRINT "VELOCITY (CM/SEC)=",X(140)
1290 REMCOPY
1300 HOME
1310 END
```

DISTRIBUTION FOR MERADCOM REPORT 2387

No. Copies	Addressee	No. Copies	Addressee

Department of Defense

1 — Director, Technical Information
Defense Advanced Research Projects
  Agency
1400 Wilson Blvd
Arlington, VA 22209

1 — Director
Defense Nuclear Agency
ATTN: TITL
Washington, DC 20305

12 — Defense Technical Information Ctr
Cameron Station
Alexandria, VA 22314

Department of the Army

1 — Commander, HQ TRADOC
ATTN: ATEN-ME
Fort Monroe, VA 23651

1 — HQDA (DAMA-AOA-M)
Washington, DC 20310

1 — HQDA (DALO-TSM)
Washington, DC 20310

1 — HQDA (DAEN-RDL)
Washington, DC 20314

1 — HQDA (DAEN-MPE-T)
Washington, DC 20314

1 — Commander
US Army Missile Research and
  Development Command
ATTN: DRSMI-RR
Redstone Arsenal, AL 35809

1 — Director
Army Materials Research and
  Development Command
ATTN: DRXMR-PL, Tech Lib
Watertown, MA 02172

1 — Technical Library
Chemical Systems Laboratory
Aberdeen Proving Ground, MD 21010

1 — Commander
US Army Aberdeen Proving Ground
ATTN: STEAP-MT-U (GE Branch)
Aberdeen Proving Ground, MD 21005

1 — Director
US Army Materiel Systems Analysis
  Agency
ATTN: DRXSY-CM
Aberdeen Proving Ground, MD 21005

1 — Director
US Army Materiel Systems Analysis
  Agency
ATTN: DRXSY-MP
Aberdeen Proving Ground, MD 21005

1 — Director
US Army Ballistic Research Laboratory
ATTN: DRDAR-TSD-S (STINFO)
Aberdeen Proving Ground, MD 21005

1 — Director
US Army Engineer Waterways Experiment
  Station
ATTN: Chief, Library Branch
      Tech Info Ctr
Vicksburg, MS 39180

1 — Commander
US Army Armament R&D Command
ATTN: DRDAR-TSS No. 59
Dover, NJ 07801

No. Copies	Addressee	No. Copies	Addressee

1	Commander
	US Army Troop Support and
	  Aviation Materiel Readiness
	  Command
	ATTN: DRSTS-MES (1)
	4300 Goodfellow Blvd
	St. Louis, MO 63120
		1	Commander
			Rock Island Arsenal
			ATTN: SARRI-LPL
			Rock Island, IL 61201

		1	HQDA ODCSLOG DALO-TSE Room 1E588 Pentagon, Washington, DC 20310
2	Director Petrol & Fld Svc Dept US Army Quartermaster School Fort Lee, VA 23801	1	Plastics Technical Evaluation Center ARRADCOM, Bldg 3401 ATTN: A.M. Anzalone Dover, NJ 07801
1	Commander US Army Electronics Research and Development Command Tech Library Division ATTN: DELSD-L Fort Monmouth, NJ 07703	1	Commander Frankford Arsenal ATTN: Library, K2400, B151-2 Philadelphia, PA 19137
1	President US Army Aviation Test Board ATTN: STEBG-PO Fort Rucker, AL 36360	1	Commandant US Army Engineer School ATTN: ATZA-CDD Fort Belvoir, VA 22060
1	US Army Aviation School Library P.O. Drawer O Fort Rucker, AL 36360	1	President US Army Airborne, Communications and Electronics
2	HQ, 193D Infantry Brigade (Pan) ATTN: AFZU-FE APO Miami 34004		ATTN: STEBF-ABTD Fort Bragg, NC 28307
2	Special Forces Detachment, Europe ATTN: PBO APO New York 09050	1	Commander Headquarters, 39th Engineer Battalion (Cbt) Fort Devens, MA 01433
2	Engineer Representative USA Research & Standardization Group (Europe) Box 65 FPO 09510	1	President US Army Armor and Engineer Board ATTN: ATZK-AE-PD-E Fort Knox, KY 40121

No. Copies	Addressee	No. Copies	Addressee
1C	Commander and Director USAFESA ATTN: FESA-TS Fort Belvoir, VA 22060	5 20 3 3	Combined Arms Support Lab, DRDME-X Sensor Technology Div, DRDME-XS Tech Reports Ofc, DRDME-WP Security Ofc (for liaison officers), DRDME-S
1	Director US Army TRADOC Systems Analysis Activity ATTN: ATAA-SL (Tech Lib) White Sands Missile Range, NM 88002	2 1 1 1	Tech Library, DRDME-WC Program Support, DRDME-HR Public Affairs Ofc, DRDME-I Ofc of Chief Counsel, DRDME-L

Department of the Navy

1	HQ, USAEUR & Seventh Army Deputy Chief of Staff, Engineer ATTN: AEAEN-MT-P APO New York 09403	1	Director, Physics Program (421) Office of Naval Research Arlington, VA 22217
1	HQ, USAEUR & Seventh Army Deputy Chief of Staff, Operations ATTN: AEAGC-FMD APO New York 09403	2	Commander, Naval Facilities Engineering Command Department of the Navy ATTN: Code 032-B; 062 200 Stovall Street Alexandria, VA 22332

MERADCOM

1	Commander, DRDME-Z Tech Dir, DRDME-ZT Assoc Tech Dir, DRDME-H Chief Engineer, DRDME-HE Chief Scientist, DRDME-HS Program Planning, DRDME-HP Program Support, DRDME-HP Systems Analysis, DRDME-HA CIRCULATE	1	US Naval Oceanographic Office Navy Library/NSTL Station Bay St. Louis, MS 39522

1	Chief, Engineer Support Lab, DRDME-N
	Chief, Engineer Service Support Lab,
	  DRDME-E
	Chief, Logistics Support Lab, DRDME-G
	Chief, Matl, Fuels, and Lubricants Lab,
	  DRDME-V
	Director, Product A&T Directorate,
	  DRDME-T
	CIRCULATE
		1	Library (Code L08A)
			Civil Engineering Laboratory
			Naval Construction Battalion Centter
			Port Hueneme, CA 93043

		1	Director Earth Physics Program Code 464 Office of Naval Research Arlington, VA 22217
		1	Naval Training Equipment Center ATTN: Technical Library Orlando, FL 32813

No. Copies	Addressee

Department of the Air Force

1	HQ USAF/RDPT ATTN: Mr. Allan Eaffy Washington, DC 20330

1	HQ USAF/LEEEU Chief, Utilities Branch Washington, DC 20330

1	US Air Force HQ Air Force Engineering & Svcs Ctr Technical Library FL 7050 Tyndall AFB, FL 32403

1	Chief, Lubrication Br Fuels & Lubrication Div ATTN: AFWAL/POSL Wright-Patterson AFB, OH 45433

1	Department of Transportation Library, FOB 10A, M494-6 800 Independence Ave, SW Washington, DC 20591

Others

1	Professor Raymond R. Fox School of Engineering & Applied Science George Washington University Washington, DC 20052

1	Reliability Analysis Center Rome Air Development Center RADC/RBRAC (I.L. Krulac) Griffiss AFB, 13441

END

FILMED

12-83

DTIC